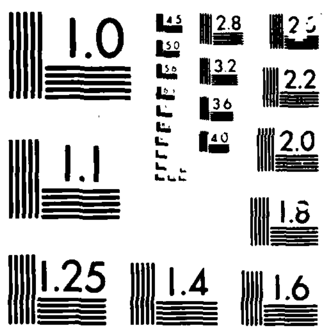
AD-A167 450 COMPUTER ASSISTED GAS CHROMATOGRAPHY FOR THERMODYNAMIC MEASUREMENTS OF POLYMER SURFACES(U) MATERIALS RESEARCH LABS ASCOT VALE (AUSTRALIA) P J CHAPPELL DEC 85 1/1

UNCLASSIFIED MRL-R-982 F/G 7/4 NL



MICROCOP' CHART

MRL-R-982 AR-004-350 (12)

AD-A167 450

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

MATERIALS RESEARCH LABORATORIES

MELBOURNE, VICTORIA

REPORT

MRL-R-982

COMPUTER ASSISTED GAS CHROMATOGRAPHY FOR
THERMODYNAMIC MEASUREMENTS OF POLYMER SURFACES

P.J.C. Chappell

Approved for Public Release

86 5 5 008

DEPARTMENT OF DEFENCE

MATERIALS RESEARCH LABORATORIES

REPORT

MRL-R-982

COMPUTER ASSISTED GAS CHROMATOGRAPHY FOR THERMODYNAMIC MEASUREMENTS OF POLYMER SURFACES

P.J.C. Chappell

ABSTRACT

The requirements for accurate measurements of thermodynamic properties of polymer surfaces using gas chromatography are detailed. Modifications to an existing chromatograph and multichannel analyser to allow digital acquisition and analysis of data are described. Preliminary results demonstrate the utility of the resulting system.

POSTAL ADDRESS: Director, Materials Research Laboratories
P.O. Box 50, Ascot Vale, Victoria 3032, Australia

DOCUMENT CONTROL DATA SHEET

REPORT NO.	AR NO.	REPORT SECURITY CLASSIFICATION
MRL-R-982	AR-004-350	Unclassified

TITLE

Computer assisted gas chromatography for thermodynamic measurements of polymer surfaces

AUTHOR(S)

P.J.C. Chappell

CORPORATE AUTHOR

Materials Research Laboratories
P.O. Box 50,
Ascot Vale, Victoria 3032

REPORT DATE	TASK NO.	SPONSOR
December 1985	DST 84/219	DSTO
FILE NO.	**REFERENCES**	**PAGES**
G6/4/8-3137	4	33

CLASSIFICATION/LIMITATION REVIEW DATE

CLASSIFICATION/RELEASE AUTHORITY
Superintendent, MRL
Organic Chemistry Division

SECONDARY DISTRIBUTION

Approved for Public Release

ANNOUNCEMENT

Announcement of this report is unlimited

KEYWORDS

Gas chromatography
Surface thermodynamics
Kevlar
Fibers
Adsorption
Inverse

COSATI GROUPS 0704

ABSTRACT

The requirements for accurate measurements of thermodynamic properties of polymer surfaces using gas chromatography are detailed. Modifications to an existing chromatograph and multichannel analyser to allow digital acquisition and analysis of data are described. Preliminary results demonstrate the utility of the resulting system.

CONTENTS

COMPUTER ASSISTED GAS CHROMATOGRAPHY FOR THERMODYNAMIC MEASUREMENTS OF POLYMER SURFACES

1. INTRODUCTION

Accurate measurement of the surface thermodynamics of polymers is essential if one is to understand how polymeric fibres such as Kevlar interact with the matrix in fibre reinforced composites. The term "surface thermodynamics" includes quantities such as enthalpies of adsorption, entropies of adsorption, adsorption isotherms, surface areas and spreading pressures. Gas chromatography lends itself to the determination of these quantities because :

(i) it can be carried out at low concentrations of adsorbate (the infinite dilution regime),

(ii) it does not need to be carried out under a vacuum,

and (iii) its accuracy is as good as, or better than, the accuracy of alternate methods, e.g. calorimetry.

The technique used for measuring the surface thermodynamics of polymer fibres involves packing the chromatograph column with the fibres to be investigated and accurately determining the retention behaviour of various probe molecules (solutes) passed through the column in the carrier gas stream. Because this is the inverse of the analytical chromatography experiment, in which the gas is being analysed, the technique is known as inverse gas chromatography (IGC). The accuracy required of the measurements depends upon the application; high accuracy is vital for studies in which small changes in surface properties, e.g. as a result of surface treatment, are being sought.

This report documents the development of a system, based on an existing gas chromatograph and multichannel analyser, for digital acquisition and analysis of chromatographic data. A brief outline of the theory is followed by a section detailing specific requirements. The bulk of the report, describing the development of the system, follows, with some results illustrating its capabilities. Finally, an extensive set of appendices gives technical details.

2. THEORY

In the infinite dilution regime, where the quantity of gas adsorbed on the substrate is proportional to its partial pressure above the substrate, the critical parameter in gas chromatography is the retention volume, V_N, given by :

$$V_N = jF\,(t_R - t_M)$$

where F is the carrier gas flow rate
t_R is the retention time of the peak
t_M is the retention time of a non-sorbed solute
and j is a correction for pressure drop across the column.

t_M is known as the gas holdup time, and $Ft_m = V_m$ is the dead volume, which is equal to the space in the column not occupied by the packing material. In the infinite dilution regime, the retention time t_R, for a solute is given simply by the first moment with respect to time of the elution profile [1].

From the retention volumes of a particular solute on a given substrate at different temperatures the isosteric heat of adsorption, ΔH_{st}, may be calculated. This quantity is the heat adsorbed when one mole of adsorbate (solute) is adsorbed by an infinite amount of substrate without a change in the fraction of surface area covered by adsorbate. ΔH_{st} is given by :

$$\left(\frac{\partial (V_N/T)}{\partial (1/T)}\right)_v = \frac{\Delta H}{R}\,st$$

where T is the temperature in K
R is the universal gas constant.

At higher concentrations (the finite concentration regime) the quantity of solute does not generally depend linearly upon the partial pressure of the solute, but rather in a more complex manner described by the adsorption isotherm. In this regime the shape of the elution profile depends upon the adsorption isotherm. Although some theoretical work has been done on non-linear isotherms [2], the problem of determining the adsorption isotherm from the elution profile is generally considered to be intractable unless special criteria are met [3]. This is certainly true of analogue data, but work in progress has demonstrated that the problem is susceptible to treatment by an iterative process if digital data are available.

Once an accurate adsorption isotherm has been determined, the surface area may be calculated via the BET (Brunauer-Emmett-Teller) [4] formalism. In this formalism the isotherm data are fitted to the formula

$$\frac{1}{q} \cdot \frac{x}{1-x} = \frac{c-1}{cq_m} x + \frac{1}{cq_m}$$

over the range $0.05 < x < 0.35$

where x = p/p_o, the partial pressure of solute relative to its saturation partial pressure

q = quantity of gas adsorbed (moles/gram of substrate)

q_m and c are parameters derived from the fit.

q_m is the value of q at monolayer coverage. c describes the temperature dependence of the adsorption. From q_m, the surface area of the substrate may be determined if the cross sectional area of the solute molecule is known :

$$A_s = q_m N_A \sigma$$

where N_A = Avogadros constant
σ = cross sectional area of solute molecule.

It is the determination of quantities such as heat of adsorption, adsorption isotherms and surface areas which are important in the thermodynamic characterisation of polymer surfaces.

3. REQUIREMENTS

Various factors impinge upon the accuracy of measurements made using gas chromatography. The two most critical are temperature and flow rate. The retention volume for a solute on a particular substrate depends linearly upon the flow rate and in a more complicated fashion upon the temperature. It is important that these two quantities be monitored and controlled to within 0.5% and 0.1°C respectively. Until recently, the other major sources of error were the physical acquisition and analysis of data; all data were recorded on a chart recorder and analysed by hand. The accuracy was only as good as the eye and hand of the person carrying out the measurements. Newer gas chromatographs offer digital signal acquisition and processing (at a cost) and allow much more rapid and accurate analysis of data. As noted above, the calculation of adsorption isotherms from elution profiles is difficult unless the data are available in digital form. Any serious attempt, then, to measure the thermodynamic properties of polymer surfaces requires digital data acquisition and analysis.

The system which has been developed to meet the above requirements is based on a Varian Series 1400 gas chromatograph with a flame ionisation

detector, and a Tracor Northern 1710 multichannel analyser (TN-1710). The modifications to the gas chromatograph were relatively straightforward and are described first. The modifications to the TN-1710 have been substantial and are described in the body of the report. Technical details are given in the Appendices.

4. GAS CHROMATOGRAPH

Temperature control in the chromatograph column is usually achieved with an oven. However, this is useful only at elevated temperatures, and there can be temperature gradients in the chromatograph oven. These problems have been overcome by pumping fluid from a thermostatted bath through a jacket surrounding the column. This allows accurate temperature control and measurement even at ambient temperatures.

The carrier gas flow rate is monitored by an MKS FM360 flow transducer, and controlled by a NUPRO double patterned fine metering valve. These give repeatable control and measurement to better than 0.5%. The pressure drop across the column, which affects retention behaviour to a lesser degree, is monitored by an MKS Baratron 223B differential 100 torr pressure transducer. All of the monitoring and flow control equipment has been mounted in a separate box with external leads to a digital readout. The recorder output from the gas chromatograph is used as the analog input to the TN-1710 analyser.

5. TN-1710 OVERVIEW

(i) The System as Purchased

The Tracor Northern-1710 multichannel analyser was purchased in 1981 and was intended for use with a diode array rapid scan spectrometer. The TN-1710 is a modular piece of equipment, comprising a mainframe and assorted plug-in modules and peripherals. The mainframe was purchased with 8 K of spectral data memory and 20 K of program memory. In addition, the following modules and peripherals were purchased:-

- TN-1710-21 optical spectrometer module with spectrometer attachments
- TN-1710-28 memory subgroup module
- TN-1710-31 remote control module
- TN-1710-37 sequential scan module
- TN-1710-24 disc tagwood module and controller card
- KSR 43 teletype
- DSD 440 dual 8" disk drive
- Disk based Flextran language interpreter.

The total system stood alone and allowed up to 16 individual spectra to be recorded prior to writing them on disk. The disk based Flextran language interpreter allowed programs to be written to analyse data.

(ii) TN-1710 Operating System

The TN-1710 is a very sophisticated piece of equipment, and operation may be considered at a number of levels. At the hardware level, and at the heart of the TN-1710, is a Digital Equipment Corporation (DEC), LSI 11/2 microprocessor which drives the analyser. Extensive use is made of read-only-memory (ROM) based firmware and disk based software. What follows is a discussion in general terms of how these features interact in the operation of the analyser.

The operating system for the TN-1710 is stored in ROM and is automatically loaded into program memory at power-up. Some parts of the operating system reside permanently in memory and are never disturbed. Other elements of the operating system may be overwritten and are reloaded from ROM by permanently resident routines as necessary. The mainframe and all of the plug-in modules have their own ROM which is loaded into program memory, as required, when the appropriate pushbutton or switch is activated. Many of the operations of the analyser, such as data acquisition and disk input/output, can be carried out using only the pushbuttons and switches.

In addition to executing these functions of the analyser, the operating system drives an oscilloscope display which presents the contents of the currently selected spectral data memory subgroup. This is very useful, because it allows data to be displayed as it is being acquired into spectral data memory. It is important to distinguish this from program memory, where the operating system and the Flextran interpreter and any Flextran programs reside. Because these two types of memory are completely separate, data acquisition and program execution can be carried out simultaneously. This is a very powerful feature of the TN-1710.

The structure of the TN-1710 operating system is also worthy of brief comment. The instruction set of the LSI-11/2 microprocessor contains a TRAP instruction. This is a software generated interrupt instruction which, with an appropriate handler, can be used as a single instruction subroutine call. The TN-1710 operating system consists of a small loop routine with a series of subroutines accessible through the TRAP instruction and handler. A more detailed description of the various subroutines and the TRAP handler isprovided in Appendix A. Discussion of the TRAP instruction and other LSI-11/2 instructions may be found in the DEC "Microprocessor and Memories" handbook.

In order to organise program memory space effectively, the TN-1710 makes extensive use of buffers. These are areas of memory set aside for various lists of addresses or parameters. The TN-1710 Flextran manual gives a partial list of buffers and their functions. A complete list and a description of their operation is given in Appendix B. One of the buffers is known as the executive loop list buffer. This buffer, which may be changed

from time to time, contains a list of all firmware modules to be executed. However, the subroutine which cycles through these module names executes only those whose priority is greater than, or equal to, the system priority as specified in the subroutine call. Those firmware modules not in memory are loaded from ROM as necessary. The effect of this priority system is shown by the following example. The display driver has a priority of 3. When Flextran is not loaded the system priority is 0, so all other modules in the loop list, including the display, will be executed. However, when Flextran is loaded and is executing, the priority is 5. Therefore the oscilloscope display will not be active. When Flextran is loaded but inactive, the priority is 1. A number of the front panel pushbuttons carry a priority of 0 and will therefore be inoperative.

(iii) TN-1710 Firmware and Software Routines

The routines used by the TN-1710 may be divided into three classes:

(a) Firmware routines on ROM, which are activated through the front panel pushbuttons and switches.

(b) Software routines which are called by the command string interpreter in response to a keyboard entry.

(c) Software routines (libraries) which are called by Flextran.

All software routines are stored on disk in machine language. The procedures for loading them are similar to those which are used to load programs from ROM. These procedures are discussed in detail in Appendix C. The structure of the ROM based routines is discussed in Appendix D.

The command string interpreter (CSI) is one of the routines automatically loaded when the TN-1710 is powered up. However, unlike most of the operating system, the CSI is stored on disk. The CSI issues a > prompt to the keyboard and will accept a two letter command. This two letter command is decoded (see Appendix E) and if it is valid, it is carried out by loading and executing the appropriate disk based routine. One of these disk based routines is the Flextran interpreter itself, which is loaded by typing the command FX after the prompt. Once Flextran is loaded, the other software routines may still be executed by entering the > prior to the two letter code. In this case, however, the CSI is not used - the Flextran interpreter calls the routine. The structure of the software routines accessed via the > prompt is presented in Appendix F. Most of the CSI routines are adequately described in the Tracor Northern manuals, but two >PX and >OD are not. >PX is described in Appendix F, while >OD is discussed later in conjunction with system modifications.

The Flextran libraries can only be accessed through Flextran. The libraries supplied with Flextran are adequately described in terms of their function in the TN-1710 Flextran Addendum, and are not discussed further here. Their structural characteristics are given in Appendix G.

(iv) Flextran

Flextran is a high-level programming language which Tracor Northern have developed for their multichannel analysers. It is an interpreter language which translates each line of code into immediate processor actions. This has the advantage of easy editing, but is substantially slower than a compiler based language.

Flextran is stored on disk and is loaded by the CSI. The language is described in detail in the "Flextran Instruction and Programming Manual" and the "Flextran Addendum".

The actual operation of the Flextran interpreter is very similar to the operation of the TN-1710 operating system. The Flextran interpreter uses TRAP instructions to access operating system routines and it also makes use of the EMT (emulator trap) instruction and an LSI-11/2 illegal instruction hardware trap to call internal routines. The TRAP and EMT instructions are used extensively in DEC software, which means that Flextran and the TN-1710 are incompatible with DEC software, such as RT-11.

6. MODIFICATIONS TO TN-1710 FOR GAS CHROMATOGRAPHY DATA ACQUISITION

In order to acquire and analyse digital data from a gas chromatograph it was necessary to make both hardware and software modifications to TN-1710. These are now described.

(i) Hardware Modifications

The most fundamental hardware modification has been the acquisition of a signal averaging module to replace the optical spectrometer module, which collects data into 512 channels simultaneously and is therefore not suitable for real time data acquisition. The signal averaging module converts the TN-1710 into a general purpose data station, capable of digitising real time analog data, with a dwell time ranging from 10 microseconds to 10000 seconds per channel.

The LSI-11/2 microprocessor itself has been upgraded by the addition of an EIS/FIS chip, which extends the instruction set to include integer multiply, integer divide, register shift and elementary floating point operations. These additional instructions accelerate arithmetic calculations in Flextran, thereby allowing data to be analysed more rapidly. (Flextran is an interpreter language and is considerably slower in execution than compiler based languages such as Fortran.) The 20 K of program memory supplied with the TN-1710 has also been upgraded, to the maximum of 32 K, in order to accommodate the larger Flextran programs necessary for data analysis.

Additional hardware purchased to facilitate data reduction includes a video terminal to replace the teletype, a small dot-matrix printer and a digital plotter.

(ii) Firmware Modifications

In order to carry out a number of the software modifications to be described below, it has been necessary to modify the operating system. In particular, it is desirable that certain actions be taken automatically when the system is powered up. Modifications to the operating system, however, are not easily made, because it is stored on read only memory (ROM) and cannot be changed. The only way to intervene at system power-up is to modify the command string interpreter (CSI), which is disk based. The CSI has been modified so that before it accepts any keyboard input it loads and executes another disk based software module, which makes the necessary changes to the operating system. Details of the routines involved and their functions are presented in Appendix H.

(iii) Software Modifications

This section includes both modifications to the TN-1710 supplied software and new software written for a specific purpose. The modified and new routines are written in machine code and are stored on disk in the same format as the software routines supplied with the TN-1710. They are accessed through the CSI by a two letter code.

a. Heading Routines

Under the TN-1710 operating system provision is made for one label (or heading) to be stored. This is clearly inadequate for a system in which up to sixteen chromatograms can be stored in spectral data memory concurrently. The problem has been overcome by creating an additional permanent buffer at power-up, to allow for each chromatogram to have its own individual heading. The operating system is modified so that disk read and write operations automatically include the heading from the currently selected memory subgroup. The method outlined in the TN-1710 mainframe manual for entering a heading is still valid, the heading being associated with the currently selected memory subgroup. Five routines have been written to handle headings. These are outlined in Appendix I. Headings may also be entered in a specific format through the chromatogram analysis program, which is discussed later.

b. Tagwood Commands

When Flextran is loaded the disk tagwood controller is inoperative because it has a lower operating priority than Flextran. This means that the pushbuttons cannot be used to transfer data to and from disk without temporarily disabling Flextran. This can be inconvenient, so five routines have been written to handle transfers between disk and spectral data memory.

These commands are described in Appendix J.

c. Printer Commands

The software module >OD is used to transfer output from Flextran to an ASCII device such as a printer. The description of >OD provided with the TN-1710 applies only to the original version of >OD. This has been substantially modified to facilitate printer and video terminal operation. The operation of >OD, both before and after the modifications, is described in Appendix K.

In addition to the changes to >OD, a number of utility routines have been written for the printer. These allow print type to be changed and linefeeds and formfeeds to be issued. The routines are listed in Appendix L.

d. Arithmetic Package

This is the most complicated software package, containing nine routines which add to the arithmetic utility of the TN-1710. These routines allow addition of a constant term to a memory subgroup, multiplication or division of a memory subgroup by a constant or another memory subgroup, up to 25 point Savitsky-Golay smoothing, first and second derivatives, and integration. The routines and their usage are described in Appendix M.

e. Autoplot

This is a single machine code routine which rapidly plots the contents of the display on the digital plotter. It was written because plotting using Flextran is too slow. Autoplot runs fourteen times as fast as an equivalent Flextran program. Autoplot is accessed by the command >AP n, where n is a pen number from 1 to 8.

f. Time and Date Routine

The instructions for displaying a time and date are loaded when the TN-1710 is powered up. However, the clock and display are inactive until the time and date routine, >TI is called. This routine asks for the date and then the time, before activating the clock and display. The clock is driven by counting hardware generated interrupts which occur at twice the AC line frequency. The timekeeping facility allows a time and date to be included on the printer output from chromatogram analyses. Tracor Northern also supply a clock module, the TN-1710-47, but it stores only the time and a day number, rather than the full date.

g. Diskette Utility Routines

These are supplied on the master disk and are used to generate disk catalogs and handle disk files. The >LC command lists the contents of the disk catalog on the terminal. However, this command does not use >OD to output the list, so listing on a printer is not possible with the utility routines as supplied. This problem has been overcome by the introduction of the >PC command, which will list the contents of a disk on the printer. Rules governing the use of >PC are identical to those given for >LC in the diskette controller manual.

h. Flextran and Libraries

Flextran is a very complicated set of machine language routines which have not been analysed in great depth. The only change made to Flextran has been the alteration from 72 to 132 of the default line length. That is, Flextran now issues a carriage return and linefeed after 132 characters rather than 72. This change has been made to accommodate a video terminal which can display up to 132 columns.

Of the Flextran libraries only library 41 has been changed. A new command has been added to it. Two additional libraries, 11 and 32, have been written. These libraries, and the changes made to library 41, are described in Appendix N.

(iv) Gas Chromatogram Analysis Program (GCAP)

GCAP is a very extensive Flextran program written to facilitate the analysis of gas chromatograms. It has two major functions and a number of minor ones:

Major:
1. Calculations of pressure- and temperature- corrected retention parameters for peaks in the gas chromatogram.
2. Isotherm calculation (both infinite dilution and finite concentration) and subsequent calculation of thermodynamic data.

Minor:
1. Chromatograph calibration
2. Baseline stripping
3. Chromatogram heading and parameter input
4. Printing and plotting results

GCAP is menu-driven and is generally self-explanatory during execution. In order to function correctly, GCAP requires that the chromatogram being analysed has a heading containing a number of parameters

in a given format. This heading may be entered through GCAP before the data is analysed.

GCAP may be loaded from the master disk in drive 1 by executing the following commands (after the prompt, *) when Flextran is loaded:-

```
M 18200
>OP 1
$LI 40
$LO ('GCAP')
```

GCAP occupies almost all of the memory available and takes nearly a minute to load from disk.

7. RESULTS

Preliminary studies using alkane probes on Kevlar have been carried out. Table 1 contains results for a column packed with Kevlar-49 roving and heated at 80°C until all of the water vapour had been driven off. Results for Kevlar which has been cleaned by solvent extraction and dried under vacuum and heat are presented in Table 2. The isosteric heats of adsorption of the alkanes on cleaned Kevlar are quite different from those on Kevlar which has been dried but not cleaned (Table 3).

The retention volumes presented in Tables 1 and 2 have been corrected for the effects of gas holdup (dead volume = 16.8 cm^3), and adsorption on the stainless steel column, measured separately, which contributes approximately 2% of the retention volume. The average percentage difference between pairs of corrected retention volumes is 0.43%. This is typical of the accuracy which can be achieved with a digital system.

Preliminary studies have also been made using octane at higher concentrations, but these have not yet reached a point at which they can be reported.

8. CONCLUSION

Existing equipment (a Varian Series 1400 chromatograph and Tracor Northern-1710 multichannel analyser) has been upgraded to allow accurate thermodynamic measurements to be made on polymer surfaces using inverse gas chromatography. Preliminary results for alkane probes on Kevlar with two different treatments illustrate the general usefulness of the technique, and the specific utility and accuracy of the digital data acquisition and analysis features of the system.

9. REFERENCES

1. E. Kucera, J. Chromatog. 19, 237 (1965).

2. K. de Clerk and T.S. Buys, J. Chromatog. 63, 193 (1971).

3. J.R. Conder and C.L. Young, Physicochemical Measurement by Gas Chromatography, p 75, Wiley-Interscience, New York, 1979.

4. S. Bruneuer, P.H. Emmett and E. Teller, J. Am. Chem. Soc. 60, 309 (1938).

TABLE 1 Corrected retention volume for n-alkanes on 7.139 g dried Kevlar as a function of temperature. Flow rate, 10 cm^3/minute; dead volume, 17.3 cm^3; column length 1.5 m.

Temperature	Corrected Retention Volume (cm^3)				
	n-C_6H_{14}	n-C_7H_{16}	n-C_8H_{18}	n-C_9H_{20}	n-$C_{10}H_{22}$
20°C	7.44	21.04	58.92	165.9	453.4
	7.42	21.07	58.84	165.1	452.0
30°C	5.27	14.07	37.35	97.9	254.2
	5.27	14.07	37.23	98.0	252.8
40°C	3.77	9.57	23.90	59.92	147.4
	3.77	9.57	23.95	60.08	148.7
50°C	2.79	6.72	16.07	38.09	89.6
	2.92	6.74	16.13	37.95	90.0

TABLE 2 Corrected retention volume for n-alkanes on 7.638 g solvent extracted and dried Kevlar as a function of temperature. Flow rate, 10 cm^3/minute; dead volume 16.8 cm^3; column length 1.5 m.

Temperature	Corrected Retention Volume (cm^3)				
	n-C_6H_{14}	n-C_7H_{16}	n-C_8H_{18}	n-C_9H_{20}	n-$C_{10}H_{22}$
20°C	5.98	23.07	89.12	344.9	1385.0
	5.97	22.96	89.87	341.9	
30°C	3.76	13.17	46.91	164.9	602.2
	3.78	13.12	46.67	164.2	
40°C	2.38	7.81	25.46	84.35	284.69
	2.38	7.83	25.47	83.76	283.86
50°C	1.53	4.71	14.34	44.36	137.58
	1.53	4.70	14.35	44.12	136.89

TABLE 3 Values of the isosteric heat of adsorption. ($\Delta H_{st.}$) for alkanes on dried Kevlar and solvent extracted Kevlar, calculated from retention data in Tables 1 and 2.

	ΔH_{st} (kJ/mole)				
	n-C_6H_{14}	n-C_7H_{16}	n-C_8H_{18}	n-C_9H_{20}	n-$C_{10}H_{22}$
Dried Kevlar	27.3	32.6	36.9	41.2	45.1
Solvent Extracted Kevlar	38.3	44.1	50.6	56.4	63.1

APPENDIX A

The TRAP instruction in the LSI-11/2 instruction set takes the form 104400_8 + XXX where XXX is a parameter ranging from 0 to 377_8. When a TRAP instruction is encountered, a software interrupt is generated, and control is transferred to the TRAP handler. The TRAP handler despatches control to the routine determined by the parameter extracted from the TRAP instruction. The TN-1710 operating system contains a table of permanently resident TRAP routines for parameters less than 100_8. The TRAP routines used range from 0 to 61_8.

The TRAP handler occupies memory locations $2124\text{-}2204_8$, the TRAP table occupies locations $2206\text{-}2404_8$ and the TRAP routines occupy locations from 2406_8 to about 13000_8, depending upon which front end (data acquisition) module is present. The TRAP routines control most of the automatic operations of the TN-1710. Some of the TRAP routines and their functions are:

TRAP Number	Function of Routine
3	Generates parameters from keyword for current memory subgroup
11	Loads subroutines from ROM or disk
12	Checks whether space is available for subroutine to be loaded
14	Checks whether subroutine is already loaded
21	Integer multiplication
22	Unsigned integer division
24	Generates digit strings for display
25	Reads a point from spectral data memory
26	Writes a point to spectral data memory
32	Conditionally executes routines in executive routine loop list
36	Calculates gross or nett integral for regions of interest
40	Generates region of interest intensification bit
51	Signed division routine
56	All diskette operations
57	Reads parameters with CSI commands

APPENDIX B

The TN-1710 operating system makes extensive use of software buffers for storage of tables and other parameters. These buffers are stored in a linked structure descending from the top of program memory. Each buffer is described by a buffer name comprising two ASCII bytes and a buffer length. The following is a list of permanent buffers, established at system power-up, and their functions:-

Buffer Name	Length (words)	Function
FO	264	disk controller usage
GX	96	group descriptors
RX	100	region of interest table
XX	32	executive routine loop list
VX	32	stores addresses of character and digit strings for display
CS	33	stores keyboard input
TI	15	stores time
HX	288	stores memory subgroup headings
SB	65	(function not determined)

Buffers TI and HX are generated by routines written at MRL and are not generated by the operating system originally supplied with the TN-1710.

APPENDIX C

ROM routines are characterised by a one word name, the low byte of which ranges from 200 to 377_8. When a TRAP instruction is used with a parameter in this range, the TRAP handler calls TRAP 11, which looks through the ROM modules for a routine which has a low byte name matching the parameter. If the routine is found it is loaded. Otherwise, the routine is sought on disk, and loaded if found.

The actual process for accessing the individual ROM locations involves memory addresses 177100 to 177136_8. These addresses, located on the I/O page of the LSI-11/2 memory, are associated with the ROMs in the various modules. In order to read a particular location in ROM, it is necessary to enter that address into the memory address for that module. Reading the memory address will then give the contents of the desired ROM location. The ROM routine is loaded by stepping sequentially through the ROM addresses.

APPENDIX D

ROM routines are stored in a linked structure, with each subroutine containing a pointer to the next subroutine in the same module. The first three words of a ROM module are a pointer to the next subroutine, a version number (e.g. 2E) for the ROM package, and a module number (if the ROM is in a plug in module). Note that the first section of code in any ROM does not have a module name. The pointer points to the next routine name. Each subsequent routine name is associated with two other words which give the length of the routine in words, and a pointer to the name of the next routine.

The module name is of the form $20000_8 + P \times 400_8 + XXX$ where P ranges from 0 to 7 and XXX ranges from 200 to 377_8. P is the priority at which the routine operates, and XXX is the byte used for identifying the routine. The length of the routine may be given as a positive or negative number. This appears to be related to whether the routine is allowed to "bump" other routines, i.e. eliminate other routines from memory in order to have enough space to load.

APPENDIX E

Routines accessed by the command string interpreter (CSI) are stored on disk as files. The file number takes the form 10000_8 + XX or 14000_8 + XX, where XX is a number ranging from 0 to 77_8. The CSI commands have two letters, sometimes followed by parameters. The two letters are decoded into two actual digits according to the following table:-

Letters	Number
ABC	0
DEF	1
GHI	2
JKL	3
MNO	4
PQRS	5
TUV	6
WXYZ	7

Thus, for example, >OD will cause file number 10041 to be loaded and executed, as would >OE, >NF etc. All CSI routines are coded in this way. Note, however, that three of the numerical combinations, 01, 23 and 77 are used by other system routines and should not be accessed through the CSI.

APPENDIX F

Because the command string interpreter (CSI) routines are stored on disk it is possible to write additional routines. In order to do this it is necessary to enter the routine into memory through the keyboard and then use the routine >PX to write the new routine on disk. To do this successfully requires a knowledge of the structure of CSI routine.

Some of the CSI routines share common code and for this reason they are stored in groups, or packages. When one of the routines is called, the entire package is loaded. The first routine in the package is numbered 10000_8 + XX, where XX is the octal code associated with the command (see Appendix E). All other routines in the package have numbers 14000_8 + XX associated with them. For example, >WU, which is part of the disk utility package, has the number 14076 associated with it.

Each routine in a package has a routine number and a pointer to the next routine. In addition, the first word of the entire package gives the total number of the bytes in that package. This number is negative for most of the CSI routines, but is positive for a few. This difference probably determines whether the routine can "bump" other routines. It is important, too, that the first few instructions are properly handled. They should follow one of the two outlines below:

A.	177XXX	number of bytes (negative)
	100XX	routine name
	XXX	pointer to next routine
	401	skip 1 instruction - normal execution begins here
	2	return - executive polling comes here
	code	code for routine
B.	177XXX	number of bytes (negative)
	100XX	routine name
	XXX	pointer to next routine
	404	skip 4 instructions
	62716	executive polling executes these
	2	4 instructions
	5722	
	2	
	code	code for routine

The difference between these two formats lies in the way the "busy bit" is handled. The "busy bit" is actually bit 15 of the routine name. When the routine is busy, bit 15 in the routine name is set to 1. This prevents the module from being bumped while it is executing. The busy bit is usually set from within the routine. At some later stage the executive will poll the various routines, and in outline A, the executive will clear the busy bit if it has been set. In outline B, the additional code prevents the executive from clearing the busy bit.

Once a new routine has been entered into memory, it is easily stored on disk using >PX. This routine asks for the starting address. This is the address of the first instruction to be stored, 177XXX in both outlines A and B. A label is then requested. After this input, >PX proceeds to write the routine(s) on disk.

When the routine or group of routines is loaded for execution, the heading and the number of bytes are not loaded. The first instruction loaded into memory is the routine name. All routines in memory are linked by pointers. This facilitates searching for particular routines.

APPENDIX G

Flextran libraries are stored on disk as files with numbers 10100_8 + XX, where XX is the library number. Thus, 10150_8 is library 40 (decimal). The libraries are structured similarly to the command string interpreter routines (see Appendix F). The library routines are divided into a section of code which is executed when the library is loaded, and the additional instructions which link into Flextran. The first few instructions of the library should follow one of the models in Appendix F. The code executed initially is used to set the busy bit and carry out other initialisations. After this code come the additional Flextran instructions. Each instruction has a two byte name, corresponding to the command used to access the instruction, followed by a pointer to the next Flextran instruction, followed by the actual code which executes the instruction. The final instruction has a pointer of 100000_8. The standard Flextran routines are linked in the same fashion to each other and to the library routines, so that when a Flextran command is given, Flextran searches through its entire list of instructions including any linked libraries.

When Flextran libraries are linked, the busy bit is set. When the libraries are delinked, the busy bit is cleared, allowing another routine to occupy the space if necessary. Flushing the libraries actually reallocates memory space so that the library routines are deleted.

APPENDIX H

The command string interpreter (CSI) is a disk based routine which is loaded by the TN-1710 operating system at power-up. It then resides permanently in memory. Modification of the CSI stored on disk is the only way to alter the operating system at power-up.

The CSI has been modified so that when it is loaded it calls routine 250. This disk-based routine carries out a number of changes to the operating system and then restores the CSI in memory to the code originally supplied, before being eliminated itself. The only evidence that routine 250 has been loaded are the changes it has made to the operating system.

Routine 250 carries out the following functions:

(i) It establishes the permanent buffers HX and TI for the heading and timekeeping routines.

(ii) It determines which front end (data acquisition) module is present and sets the data file type accordingly. For example, the diode array module generates file type 13, while the signal averager generates file type 16.

(iii) It loads and locks in routine >OD for later use with Flextran.

(iv) It loads and locks in routine 251, whose functions are described below, and adds it to the executive loop list.

(v) It checks whether the extended instruction/floating instruction chip is present, and if so it makes minor changes to the operating system.

(vi) It restores the CSI as noted above.

Module 251 carries out the following functions:

(i) It generates the correct address for the spectral data memory heading display

(ii) It makes changes to the operating system so that the tagwood package (see Appendix J) functions properly

(iii) It operates the time and date display after the initial parameters have been installed.

APPENDIX I

The five routines in the spectral data memory heading package are listed below with their functions:-

Command	Function
>IH	Input heading into currently selected memory subgroup
>CH	Copy current heading to another subgroup
>LH	List all headings
>DH	Delete current heading
>XH	Delete all headings

None of these commands requires any parameters.

APPENDIX J

The five tagword routines and their functions are:-

Command	Function
>TR X Y	Read data from file X on drive Y. If Y is omitted, use drive 2. If X is omitted, the file number defaults to one greater than the last file read (the read, write and delete tagwords are automatically incremented after their respective commands are executed).
>TW X Y	Write data to file X on drive Y. Defaults as for >TR.
>TD X Y	Delete data file X on drive Y. Defaults as for >TR.
>TL	List the read, write and delete tagwords.
>TT n	Change data file type to n where $13 \leq n \leq 17$.

APPENDIX K

The version of >OD supplied with the TN-1710 allowed a device other than the terminal to be selected for Flextran output. In order for output via >OD to occur, it was necessary to lock it in below Flextran's own output routine. The usage of >OD was:-

>OD n	n = 0	disables >OD, allowing it to be bumped. Flextran's own output routine is used
	n = 1	output to line printer
	n = 2	output to paper tape punch
	n = 3	output to EIA device

>OD has been revised and the current usage is:

>OD n	n = 0,1,2, as above	
	n = 3	output to terminal through >OD routine
	n = 4	output to EIA device

When >OD is automatically loaded by routine 250 at power-up option 3 is selected. This option directs Flextran output to the terminal via >OD rather than through Flextran's own output routine. This allows for the correct handling of deleted characters (backspace, space, backspace) on the video terminal. Flextran's own output routine is written for a teleprinter and uses \ to indicate a deleted character.

In order to use the printer, which is connected to the TN-1710 through the EIA board, it is necessary to enter >OD 4. All the output, including keyboard echo, then goes to the printer. >OD 3 restores output to the terminal.

APPENDIX L

The six printer routines are described below:-

Command	Function
>NC n	Printer set for normal characters, n = 9,10,12 or 17 characters per inch
>BO n	Printer set for bold characters. n as for >NC
>LQ n	Printer set for letter quality characters. n as for >NC
>FP n	Feed paper by n lines. If n is omitted, a formfeed is given
>PL	These routines "lock" and "unlock" the printer.
>PU	They do not affect printer operation with >OD. The printer occasionally generates spontaneous interrupts which can halt data acquisition. "Locking" the printer avoids this problem, but prevents pushbutton output to the printer. The printer must be "unlocked" before pushbutton output to the printer (by setting the selector to EIA) can take place.

APPENDIX M

The format and usage of the commands in the arithmetic package are presented below. Those involving more than one memory subgroup ask for the other subgroup to be selected and any key then to be pressed.

Command	Function
>AK n	Adds a constant n to the currently selected memory subgroup. n may be positive or negative. If negative, there must be no space between the K and -.
>MK x	Multiplies the currently selected memory subgroup by x, where x is a floating point number. If x is omitted, a value of 1 is assumed.
>DK x	Divides the currently selected memory subgroup by x. Default as for >MK x.
>MM x	Multiply the currently selected memory subgroup by another memory subgroup of the same size, and divide the result by x. Default as for >MK x.
>DM x	Divide the currently selected memory subgroup by another memory subgroup of the same size, and multiply the result by x. Default as for >MK x.
>SM x	Process the currently selected memory subgroup through an x-point Savitsky-Golay smoothing routine. $3 \leq x \leq 25$, x odd.
>FD x	Process the currently selected memory subgroup through an x-point Savitsky-Golay first derivative routine. x as for >SM x.
>SD x	Process the currently selected memory subgroup through an x-point Savitsky-Golay second derivative routine. x as for >SM x.
>IN	Integrate currently selected memory subgroup.

APPENDIX N

Two additional libraries, 11 and 32, have been written for Flextran. Library 11 handles terminal type and line length. It comprises two commands:-

$VT n	Set terminal type to video terminal (backspace - space - backspace delete) with n characters per line.
$TT n	Set terminal type to teleprinter (backslash delete) with n characters per line.

Library 32 is a printer package, comprising four commands:

$LF n	If n = 0, formfeed on printer If n > 0, n linefeeds on printer If n < 0, set line spacing to n/144 inches
$CD n	Set to 9,10,12 or 17 characters per inch
$PM n,dd	Select print mode n = 0 or 1 (cancel or select mode dd) dd = LF, include linefeed with carriage return DD, print double density characters BO, print bold characters LQ, print letter quality characters UL, underline characters EX, print expanded characters
$DI n,str,m	Print dot image data n = 0 (single density) or 1 (double density) str = string variable containing data m = number of points to be printed from string

One additional command has been added to library 41, which reads and writes Flextran data files. The file type for these operations is 4. The command $FT n allows the file type to be set to n, where $0 < n < 20_8$.

END
FILMED

6-86

DTIC